МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

ИНСТИТУТ НЕПРЕРЫВНОГО И ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ (№44)

ОЦЕНКА

ПРЕПОДАВАТЕЛЬ

Старший преподаватель		А. В. Аксенов
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №4

Создание представлений на основании запросов на выборку в СУБД SQLite. Запросы на модификацию и удаление данных в СУБД SQLite

по дисциплине: Базы данных

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №	B6441		А. А. Сергеев
	номер группы	подпись, дата	инициалы, фамилия

Студенческий билет № 2013/0875

Санкт-Петербург 2019

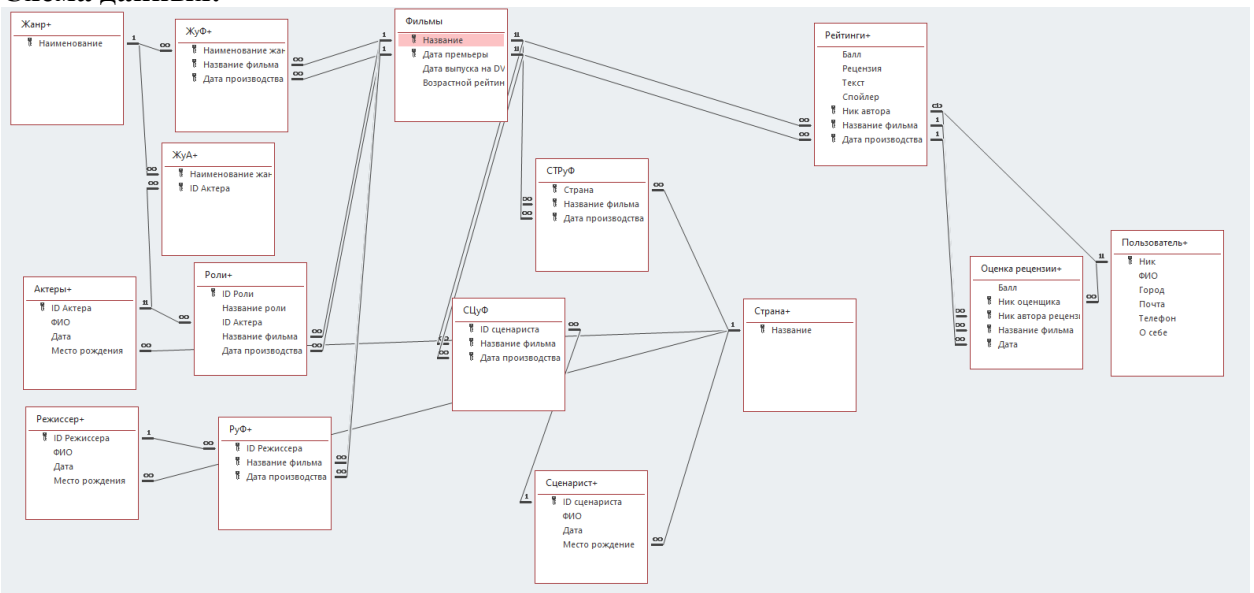
Цель работы:

Освоение синтаксиса операторов SELECT, UPDATE и DELETE при выполнении запросов на выборку, модификацию и удаление данных. Приобретение навыков использования средств языка SQL для выполнения типовых запросов к данным в реляционной модели. Знакомство с принципами работы представлений в реляционных СУБД.

Задание:

Реализовать схему данных, полученную в ЛР1, в виде скрипта SQL в СУБД SQLite. База данных должна содержать ограничения целостности, указанные в ЛР2. Индивидуальное задание отсутствует.

Схема данных:



Листинг:

```
PRAGMA foreign_keys=on;
/*
DROP TABLE IF EXISTS Student;
DROP TABLE IF EXISTS Gruppa;

CREATE TABLE Gruppa (
Number VARCHAR(8) PRIMARY KEY,
Cathedra INTEGER NOT NULL
);

CREATE TABLE Student (
FIO VARCHAR(100) NOT NULL,
ZachetkaNum INTEGER PRIMARY KEY,
GroupNumber VARCHAR(6) NOT NULL REFERENCES Gruppa(Number)
);

INSERT INTO Gruppa VALUES ("B6441", 44);
INSERT INTO Gruppa VALUES ("B7441", 44);
INSERT INTO Student VALUES ("Bill", 123, "B6441");
INSERT INTO Student VALUES ("Bill", 124, "B7441");
*/
```

```sql
DROP TABLE IF EXISTS CF;
DROP TABLE IF EXISTS GF;
DROP TABLE IF EXISTS GA;
DROP TABLE IF EXISTS PF;
DROP TABLE IF EXISTS SF;
DROP TABLE IF EXISTS Producers;
DROP TABLE IF EXISTS Scenarists;
DROP TABLE IF EXISTS ReviewRating;
DROP TABLE IF EXISTS Reviews;
DROP TABLE IF EXISTS Users;
DROP TABLE IF EXISTS Roles;
DROP TABLE IF EXISTS Actors;
DROP TABLE IF EXISTS Films;
DROP TABLE IF EXISTS Countries;
DROP TABLE IF EXISTS Genres;


CREATE TABLE Films (
Title VARCHAR(100) NOT NULL,
ReleaseDate DATE NOT NULL,
RetailDate DATE Check (ReleaseDate<RetailDate),
Age VARCHAR(3),
primary key(Title, ReleaseDate)
);

CREATE TABLE Genres (
Genre VARCHAR(15) PRIMARY KEY
);

CREATE TABLE Actors (
ActorID numeric PRIMARY KEY,
FIO VARCHAR(50),
Bdate Date,
Country varchar(50) NOT NULL REFERENCES Countries(Country)
);

Create Table Countries (
Country varchar(50) primary key
);

Create table Roles (
RoleID numeric primary key,
RoleName varchar(50) not null,
ActorID numeric NOT NULL REFERENCES Actors(ActorID),
FilmTitle varchar(100) NOT NULL,
ReleaseDate DATE NOT NULL,
Foreign Key (FilmTitle, ReleaseDate) references Films(Title, ReleaseDate)
);

Create table Producers (
ProducerID numeric primary key,
```

```
FIO varchar(50) not null,
Bdate Date,
Country varchar(50) NOT NULL REFERENCES Countries(Country)
);

Create table Scenarists (
ScenaristID numeric primary key,
FIO varchar(50) not null,
Bdate date,
Country varchar(50) NOT NULL REFERENCES Countries(Country)
);

Create Table Reviews (
Score decimal (1, 1) not null CHECK(Score between 1 and 10),
Review boolean not null,
ReviewText text,
ReviewDate date,
Spoiler boolean,
Author varchar(50) NOT NULL REFERENCES Users(Nickname),
FilmTitle varchar(100) NOT NULL,
ReleaseDate DATE NOT NULL,
primary key(Author, FilmTitle, ReleaseDate)
Foreign Key (FilmTitle, ReleaseDate) references Films(Title, ReleaseDate)
);

Create Table Users (
Nickname varchar(50) primary key,
FIO varchar(50) not null,
City varchar(50),
Email varchar(50),
Mafon varchar(15),
About text
);

Create Table ReviewRating (
Score decimal (1, 1) not null CHECK(Score between 1 and 5),
RaterNick varchar(50) NOT NULL REFERENCES Users(Nickname),
ReviewAuthorNick varchar(50) NOT NULL,
FilmTitle varchar(100) NOT NULL,
ReleaseDate date NOT NULL,
primary key(RaterNick, ReviewAuthorNick, FilmTitle, ReleaseDate),
Foreign Key (ReviewAuthorNick, FilmTitle, ReleaseDate) references Reviews(Author,
 FilmTitle, ReleaseDate),
check (RaterNick<>ReviewAuthorNick)
);

Create Table GF (
Genre varchar(15) NOT NULL REFERENCES Genres(Genre),
FilmTitle varchar(100) NOT NULL,
ReleaseDate DATE NOT NULL,
Foreign Key (FilmTitle, ReleaseDate) references Films(Title, ReleaseDate),
```

```
primary key(Genre, FilmTitle, ReleaseDate)
);

Create Table GA (
Genre varchar(15) NOT NULL REFERENCES Genres(Genre),
Aid numeric NOT NULL REFERENCES Actors(ActorID),
primary key(Genre, Aid)
);

Create Table PF (
ProducerID numeric NOT NULL REFERENCES Producers(ProducerID),
FilmTitle varchar(100) NOT NULL,
ReleaseDate DATE NOT NULL,
Foreign Key (FilmTitle, ReleaseDate) references Films(Title, ReleaseDate),
primary key(ProducerID, FilmTitle, ReleaseDate)
);

Create Table SF (
ScenaristID numeric not null  references Scenarists(ScenaristID),
FilmTitle varchar(100) NOT NULL,
ReleaseDate DATE NOT NULL,
Foreign Key (FilmTitle, ReleaseDate) references Films(Title, ReleaseDate),
primary key(ScenaristID, FilmTitle, ReleaseDate)
);

Create Table CF (
CountryName varchar(50) not null references Countries(Country),
FilmTitle varchar(100) NOT NULL,
ReleaseDate DATE NOT NULL,
Foreign Key (FilmTitle, ReleaseDate) references Films(Title, ReleaseDate),
primary key(CountryName, FilmTitle, ReleaseDate)
);



INSERT INTO Films VALUES ("Visage", "2017-08-12", "2017-09-12", "12+");
INSERT INTO Films VALUES ("Torn", "2017-08-15", "2017-09-15", "18+");
INSERT INTO Films VALUES ("Stone", "2013-05-16", "2013-06-16", "12+");
INSERT INTO Films VALUES ("Snake Awakening", "2011-05-10", "2011-06-10", "3+");
INSERT INTO Films VALUES ("Silver Shroud", "2010-03-10", "2010-04-10", "12+");

INSERT INTO Genres VALUES ("Horror");
INSERT INTO Genres VALUES ("Action");
INSERT INTO Genres VALUES ("Drama");
INSERT INTO Genres VALUES ("Fantasy");
INSERT INTO Genres VALUES ("Sci-Fi");

INSERT INTO Countries VALUES ("Russia");
INSERT INTO Countries VALUES ("Germany");
INSERT INTO Countries VALUES ("Spain");
INSERT INTO Countries VALUES ("Italy");
```

```
INSERT INTO Countries VALUES ("Poland");

INSERT INTO Actors VALUES (1 ,"Veselchakov Andrey Vadimovich","1991-07-
12", "Russia");
INSERT INTO Actors VALUES (2 ,"Swine Till Stiger","1985-03-10", "Germany");
INSERT INTO Actors VALUES (3 ,"Kassas Mario Da silva","1983-05-13", "Spain");
INSERT INTO Actors VALUES (4 ,"Celentano Adriano Zidaniovich","1980-09-
10", "Italy");

INSERT INTO Roles VALUES (1,"Bandit" ,1,"Visage", "2017-08-12");
INSERT INTO Roles VALUES (2,"Doctor" ,1,"Visage", "2017-08-12");
INSERT INTO Roles VALUES (3, "Exploder" ,2,"Stone", "2013-05-16");
INSERT INTO Roles VALUES (4, "Doctor" ,3,"Snake Awakening", "2011-05-10");
INSERT INTO Roles VALUES (5, "Diffuser" ,4,"Silver Shroud", "2010-03-10");

INSERT INTO Producers VALUES (1, "Steven Spielberg", "1967-04-12", "Russia");
INSERT INTO Producers VALUES (2, "Ridley Scott", "1973-05-11", "Spain");
INSERT INTO Producers VALUES (3, "Tony Scott", "1955-05-05", "Italy");
INSERT INTO Producers VALUES (4, "Roland Emmerich", "1956-06-06", "Germany");
INSERT INTO Producers VALUES (5, "Peter Jackson", "1973-05-11", "Poland");

INSERT INTO Scenarists VALUES (1, "Neill Blomkamp", "1973-05-11", "Poland");
INSERT INTO Scenarists VALUES (2, "Tim Burton", "1983-03-13", "Spain");
INSERT INTO Scenarists VALUES (3, "Tom Holland", "1953-05-11", "Russia");
INSERT INTO Scenarists VALUES (4, "Joe Dante", "1993-05-11", "Italy");
INSERT INTO Scenarists VALUES (5, "Stephen Daldry", "1943-05-11", "Germany");

INSERT INTO Users VALUES ("qwerty", "Sergey Feodosovich",null,null,null,null);
INSERT INTO Users VALUES ("qwerty12", "Alexander Feodosovich", null, null, null,
null);
INSERT INTO Users VALUES ("Storm", "Alexey Feodosovich", null, null, null, null);
INSERT INTO Users VALUES ("Burea", "Vasily Feodosovich", null, null, null, null);

INSERT INTO Reviews VALUES (6, 1, "fsdfdsf", "2019-09-
12", 1, "qwerty", "Visage", "2017-08-12");
INSERT INTO Reviews VALUES (8, 0, null, "2019-09-
10", 1, "qwerty12", "Visage", "2017-08-12");
INSERT INTO Reviews VALUES (5, 1, "fsdfdsf", "2019-09-
02", 1, "Storm", "Silver Shroud", "2010-03-10");
INSERT INTO Reviews VALUES (9, 1, "fsdfdsf", "2019-09-
01", 1, "Burea", "Silver Shroud", "2010-03-10");
INSERT INTO Reviews VALUES (6, 1, "fsdfdsf", "2019-08-
12", 1, "qwerty12", "Snake Awakening", "2011-05-10");

INSERT INTO ReviewRating VALUES (2, "qwerty12", "qwerty", "Visage", "2017-08-
12");
INSERT INTO ReviewRating VALUES (4,  "Storm", "qwerty12", "Visage", "2017-08-
12");
INSERT INTO ReviewRating VALUES (3,  "qwerty12", "Storm", "Silver Shroud", "2010-
03-10");
```

```
INSERT INTO ReviewRating VALUES (1,  "qwerty", "Burea", "Silver Shroud", "2010-
03-10");
INSERT INTO ReviewRating VALUES (5,  "Burea", "qwerty12", "Snake Awakening", "201
1-05-10");


INSERT INTO GF VALUES ("Horror", "Silver Shroud", "2010-03-10");
INSERT INTO GF VALUES ("Action", "Silver Shroud", "2010-03-10");
INSERT INTO GF VALUES ("Drama", "Snake Awakening", "2011-05-10");
INSERT INTO GF VALUES ("Sci-Fi", "Stone", "2013-05-16");
INSERT INTO GF VALUES ("Drama", "Torn", "2017-08-15");

INSERT INTO GA VALUES ("Drama", 1);
INSERT INTO GA VALUES ("Drama", 2);
INSERT INTO GA VALUES ("Horror", 3);
INSERT INTO GA VALUES ("Sci-Fi", 2);
INSERT INTO GA VALUES ("Action", 4);

INSERT INTO PF VALUES (1, "Silver Shroud", "2010-03-10");
INSERT INTO PF VALUES (2, "Silver Shroud", "2010-03-10");
INSERT INTO PF VALUES (2, "Snake Awakening", "2011-05-10");
INSERT INTO PF VALUES (3, "Stone", "2013-05-16");
INSERT INTO PF VALUES (4, "Torn", "2017-08-15");

INSERT INTO SF VALUES (1, "Silver Shroud", "2010-03-10");
INSERT INTO SF VALUES (2, "Silver Shroud", "2010-03-10");
INSERT INTO SF VALUES (2, "Snake Awakening", "2011-05-10");
INSERT INTO SF VALUES (3, "Stone", "2013-05-16");
INSERT INTO SF VALUES (4, "Torn", "2017-08-15");

INSERT INTO CF VALUES ("Russia", "Silver Shroud", "2010-03-10");
INSERT INTO CF VALUES ("Germany", "Silver Shroud", "2010-03-10");
INSERT INTO CF VALUES ("Spain", "Snake Awakening", "2011-05-10");
INSERT INTO CF VALUES ("Italy", "Stone", "2013-05-16");
INSERT INTO CF VALUES ("Poland", "Torn", "2017-08-15");
```

Листинг запросов:

--1) Актеры, которые не играли в фильмах со средним баллом ниже 7

```
select FIO
from actors
except
select FIO--, roles.filmtitle
from reviews, actors, roles
where actors.actorid=roles.actorid
and reviews.filmtitle=roles.filmtitle
group by roles.filmtitle
having avg(score)<7
```

--2) Режжисер, который снимает в одном и том же жанре

```
select FIO, genre
from  PF, producers, gf
where
PF.producerid=producers.producerid
and pf.filmtitle=gf.filmtitle
and gf.releasedate=pf.releasedate
group by FIO
having count( distinct genre)=1
```

--3) Посчитать кол-во комедий для каждой страны, выпущенных в 2019

```
select countryname, count(genre)
from Films, cf, gf, countries
where films.title=gf.filmtitle
and cf.countryname=countries.country
and cf.filmtitle=films.title
and Genre="Com"
and films.releasedate like '%2019%'
group by country
```

--4) Увеличить на 1 балл рейтинги фильма, если в тексте рецензии есть слово «солдат» и бал рейтинга <10

```
Update reviews
set score=score+1
where ReviewText like '%soldat%' --or reviewtext like '%Soldat%')
and score <=9
```

Вывод:

В данной работе освоил синтаксис операторов SELECT, UPDATE и DELETE при выполнении запросов на выборку, модификацию и удаление данных.